

AD-A201 059

RSRE  
MEMORANDUM No. 4209

ROYAL SIGNALS & RADAR  
ESTABLISHMENT

A THEOREM CONNECTING  
ADAPTIVE FEED-FORWARD LAYERED NETWORKS  
AND NONLINEAR DISCRIMINANT ANALYSIS

Authors: A R Webb & D Lowe

PROCUREMENT EXECUTIVE,  
MINISTRY OF DEFENCE,  
RSRE MALVERN,  
WORCS.

DTIC  
ELECTE  
NOV 21 1988  
S E D

UNLIMITED

88 11 11 000

Royal Signals and Radar Establishment

Memorandum 4209

A Theorem Connecting  
Adaptive Feed-forward Layered Networks  
and  
Nonlinear Discriminant Analysis. <sup>a</sup>

A.R. Webb & David Lowe  
Speech Research Unit,  
Royal Signals and Radar Establishment  
St Andrews Road, Great Malvern,  
Worcestershire WR14 3PS, U.K.

12<sup>th</sup> August 1988

Abstract

This paper provides a theorem which illustrates why a general adaptive feed-forward layered network with linear output units can perform well as a pattern classification device. The central result is that minimising the error at the output of the network is equivalent to maximising a particular norm, the Network Cost Function, at the output of the hidden units. If the total covariance matrix is full rank and the targets are appropriately chosen, then this cost function relates the inverse of the total covariance matrix and the weighted between class covariance matrix of the hidden unit patterns. In a linear network it is shown how our theorem can reproduce the result recently obtained by Gallinari *et.al.* as a special case. We present numerical simulations to illustrate the theorem and to show that alternative choices for the cost function at the hidden layer are not maximised, generally, in a nonlinear situation.

<sup>a</sup>This work was stimulated by a conjecture of John Bridle, to whom we are also grateful for communicating the work of Gallinari *et.al.*

Copyright © Controller HMSO, London, 1988.

## Contents

1	Introduction.	1
2	Discussion of the Network.	2
3	Theoretical Analysis.	5
3.1	Particular target coding schemes. . . . .	9
3.2	The Linear Adaptive Layered Network. . . . .	11
4	Numerical Illustration.	15
5	Conclusion.	20
A	Proof of the cost function equation.	24
B	Numerical Solution of the Least-squares Problem.	25

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



## List of Figures

- 1 Plots of various cost functions as a function of iteration number for a linear network with 4 hidden units. . . . . 18
- 2 Plots of various cost functions as a function of iteration number for a non-linear network with 4 hidden units. . . . . 19
- 3 Plots of various cost functions as a function of iteration number for a non-linear network with 6 hidden units. . . . . 21

## List of Tables

- 1 Class numbers for the Contiguity problem . . . . . 16

## 1 Introduction.

Adaptive feed-forward layered networks as exemplified by the *Multi-layer Perceptron* are known to be particularly useful as pattern classification techniques (see for instance [1,2]). What is not understood is why they perform good classification, and what underlying mechanism is responsible.

In certain instances, it is possible to identify the action of a layered network structure with the operation of a conventional classification scheme. For instance, a linear Perceptron performing an auto-associative task is equivalent to a Principal Component analysis of the data [3]. This result was generalised by Gallinari *et.al.* [4] to a linear perceptron performing an hetero-associative function. This work was important because it made explicit the fact that a linear network performing a one-from- $N$  classification to minimise the total mean-square output error, did so by implicitly maximising the ratio of the determinants of the between class and total covariance matrices. If a (linear) transformation of the input data can be made which produces an adequate separation of the classes as determined by the between class and total covariance matrices, then a subsequent linear sectioning procedure should produce good classification results. Thus it is clear why the linear Multi-layer Perceptron is capable of performing well in such circumstances. However, for a more interesting *nonlinear* transformation from the input data to the (usually, though not necessarily, dimension-reducing) space spanned by the hidden units much less is known outside empirical observation. This nonlinear transformation may be the usual logistic transformation of the scalar products between input vectors and weight vectors as in the traditional multi-layer perceptron. Alternatively, it may be the nonlinear transformation of the norm of the vector difference between input data and weight vectors, as in the Radial Basis Function network [5].

This study reports theoretical and numerical results on a subclass of general layered nonlinear feed-forward adaptive networks which demonstrate why such networks have the ability to perform nonlinear discriminant analysis successfully.

The object of interest in this paper is that class of layered feed-forward network which

takes input data, performs an arbitrary nonlinear transformation to a space controlled by 'hidden' units, and finally performs a linear transformation which attempts to minimise the mean-square error to a set of known output targets. This network class will be made more explicit in the next section. By a theoretical study of this structure, it will be apparent that a good discrimination between classes in the space of the hidden units is obtained *implicitly*, by requiring a minimisation of the output error. However, the cost function maximised at the outputs of the hidden layer is not a common choice in conventional linear or nonlinear discriminant analysis (see for instance [4,6]). It will be shown that, for the linear layered network, maximisation of the proposed cost function is equivalent to the maximisation of more popular cost functions associated with discriminant analysis.

## 2 Discussion of the Network.

This section discusses the general class of adaptive feed-forward networks, and the subclass of networks relevant to the theorem proposed in section 3.

In conventional feed-forward layered networks, data in the form of patterns represented as  $n$  dimensional (real valued) vectors are mapped by a nonlinear transformation on to  $n'$  dimensional target vectors in the following fashion. The input patterns are presented to a set of  $n$  input units. Each input unit is totally connected to a set of  $n_0$  'hidden' units (hidden from direct contact with the environment). Associated with each link between the  $i$ -th unit in the input layer and the  $j$ -th unit in the hidden layer is a scalar weight value,  $\mu_{ij}$ . Usually, the fan-in to a hidden node takes the form of a hyperplane, and the input to node  $j$  is of the form  $\theta_j = \sum_{i=1}^n I_i \mu_{ij}$  where  $I_i^p$  is the  $i$ -th component of the  $p$ -th input pattern vector. In the case of the radial basis function network [5], this fan-in takes the form of a hypersphere, i.e.  $\theta_j = \sum_{i=1}^n (I_i^p - \mu_{ij})^2$  for a Euclidean vector norm. The rôle of the hidden unit is to accept the fan-in and to pass it through a (generally) nonlinear transfer function

$$\phi_j \equiv \phi(\mu_{0j} + \theta_j) \quad (1)$$

where  $\mu_{0j}$  is a local 'bias' associated with each hidden unit.

The hidden layer units are fully connected to a set of  $n'$  output units corresponding to

the components of the  $n'$  dimensional vector in the output 'target' space. The strength of the link between the  $j$ -th hidden unit and the  $k$ -th output unit is  $\lambda_{jk}$  and thus the value received at the  $k$ -th output unit is  $\Theta_k = \sum_{j=1}^{n_0} \lambda_{jk} \phi_j$ . In general, the output from the  $k$ -th output unit is a nonlinear function of its input,

$$O_k = \Phi_k(\lambda_{0k} + \Theta_k) \quad (2)$$

where  $\lambda_{0k}$  is the bias associated with the  $k$ -th output unit.

The effect of this class of networks is to produce an interpolation surface [5,2] in the high dimensional space  $\mathbb{R}^n \otimes \mathbb{R}^{n'}$  which is entirely determined once a suitable set of values for the parameters  $\{\lambda, \mu\}$  has been specified. The ability of the network subsequently to generalise depends on the shape of this interpolating surface; if the network is sufficiently complex, it will be possible to find a set of parameters which produces an interpolating surface which passes exactly through the set of training patterns. This will not be a good generalisation strategy as the noise in the data will also have been fitted. Alternatively, if the network geometry is not complex enough, there will not be a choice of parameter values which will allow the interpolating surface to represent the relationships in the training data. This paper is not concerned with generalisation, but with training, and what it means to have a suitable set of parameters conditional upon the training data.

The parameters of the network are determined conditional upon a set of  $P$  input training patterns,  $\{|I\rangle^p \in \mathbb{R}^n\}^1$  and their associated targets  $\{|T\rangle^p \in \mathbb{R}^{n'}\}$ ,  $p = 1, 2, \dots, P$ . Conventionally, the parameters are chosen so as to minimise the mean-square error between the actual output of the network and the desired target patterns, i.e. the aim is to minimise

$$E = \sum_{p=1}^P || |T\rangle^p - |O\rangle^p ||^2 \quad (3)$$

$$\equiv \sum_{p=1}^P \sum_{k=1}^{n'} \left\{ T_k^p - \Phi_k \left( \lambda_{0k} + \sum_{j=1}^{n_0} \phi_j [\mu_{0j} + \theta_j \lambda_{jk}] \right) \right\}^2$$

<sup>1</sup>In the 'bra-ket' notation for vectors, a column vector  $(z_1, z_2, \dots)$  is written as  $|z\rangle$  (the 'ket'). The corresponding row vector is denoted  $\langle z|$  (the 'bra' vector). A scalar product between  $|z\rangle$  and  $|y\rangle$  is given by  $\langle y|z\rangle$  and  $|y\rangle\langle z|$  is a linear operator with matrix elements  $y_i z_j$ .

where  $T_k^p$  is the  $k$ -th component of the  $p$ -th target pattern vector. Since this is an explicit, differentiable nonlinear function of the parameters of interest, one can use one of the many nonlinear optimisation techniques to find an acceptable local minimum [2] which will give a suitable set of weight values. Although many other error functions may be chosen to be minimised at the output of the network, we will see that this particular choice has merits for discriminant analysis.

The one slight variant on this general theme that the rest of the paper requires, is that the transfer functions on the output units are *linear*;  $\Phi_k(x) = x \forall k$ . The important consequence of this restriction, analytically and numerically, is that the weights connecting the hidden units to the output units may be analysed by *linear* optimisation methods. In particular, given the set of weights connecting the input to hidden units, the hidden-to-output units may be adjusted by a linear least-mean-squares method to produce a *global* minimum in the error subspace spanned by the set of weights  $\{\lambda_{jk}\}$ . Consequently, given this latter set of weights, the initial input-to-hidden weights may be adjusted by a nonlinear optimisation strategy to find a better local minimum in the error subspace determined by the set of weights  $\{\mu_{ij}\}$ . This procedure may continue iteratively. For every 'slow' adjustment of the input-to-hidden weights, the hidden-to-output weights respond rapidly always maintaining the global error minimum in that subspace - the output weights are 'slaved' to the behavior of the input weights (for a numerical comparison between this hybrid methodology and solving the entire set of weights by nonlinear optimisation see [7]). This hybrid method is also closely related to the solution of the radial basis function network, if the radial basis function centres (corresponding to the knots in curve fitting) are allowed to adjust themselves by nonlinear methods.

It is interesting that least-squares error minimisation and linear output units in a general feed-forward layered network illuminates the success of such structures in discrimination problems as we now demonstrate.

### 3 Theoretical Analysis.

In this section, the error which is minimised at the output of the network will be analysed to reveal the cost function that is implicitly maximised by the network at the output of the hidden units. This will be accomplished in two stages. Firstly, it will be shown that the effect of the biases at the output of the network is to ensure that the mean output of the network equals the mean target pattern. This result allows the output biases to be removed by a rescaling of the target vectors and the outputs of the hidden units. Secondly, the error in terms of the rescaled variables is analysed to obtain the actual cost function which is being maximised. Further subsections will consider the interpretation of the result in certain limiting circumstances.

The error at the output of the network introduced in the previous section may be expressed as

$$\begin{aligned} E &= \|\Delta H - T\|^2 \\ &\equiv \text{Tr}[(\Delta H - T)(H^* \Delta^* - T^*)] \end{aligned} \quad (4)$$

where  $\Delta^*$  indicates the transpose of matrix  $\Delta$  and  $\text{Tr}$  denotes the trace operation. The matrix  $\Delta$  is an  $n' \times (n_0 + 1)$  array of weight values, including the biases,

$$\Delta = \begin{bmatrix} \lambda_{01} & \lambda_{11} & \dots & \lambda_{n_0 1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{0n'} & \lambda_{1n'} & \dots & \lambda_{n_0 n'} \end{bmatrix} \quad (5)$$

Matrix  $T$  is an  $n' \times P$  array of desired 'target' values, i.e.  $P$  vectors each of length  $n'$ . For the moment, the matrix elements of the target array will be denoted  $t_{ij}$ . Subsequently, these values will be restricted to either zero, unity, or the reciprocal of the square root of the number of elements in each class. Matrix  $H$  is the  $(n_0 + 1) \times P$  array of the  $P$  output vectors of the  $n_0$  hidden units plus a unit with unity output to feed the bias weights. These output vectors at the hidden layer are treated as input vectors to the final transformation

without any *a priori* assumptions regarding their origins. Matrix  $\mathbf{H}$  may be expressed as

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \phi_1^1 & \phi_1^2 & \dots & \phi_1^P \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n_0}^1 & \phi_{n_0}^2 & \dots & \phi_{n_0}^P \end{bmatrix} \quad (6)$$

where  $\phi_j^p$ ,  $j = 1, \dots, n_0$ ,  $p = 1, \dots, P$  is the output value at the  $j$ -th hidden unit corresponding to the  $P$ -th pattern.

Note that the matrix  $(\Delta\mathbf{H})$  may be expressed in the form

$$\Delta\mathbf{H} = |\lambda_0\rangle\langle 1| + \mathbf{A}'\mathbf{H}' \quad (7)$$

where  $|\lambda_0\rangle$  denotes the bias vector over all  $n'$  output units, and  $\langle 1|$  denotes a row vector with '1' in every position. Thus,  $\mathbf{A}'$  is the array of weights in the final layer, *not* including the biases, and  $\mathbf{H}'$  is the  $n_0 \times P$  array of elements  $[\phi_j^p]$ . This nomenclature has been introduced so as to separate out the effect of the bias vector. The reason for this is that the bias vector compensates for the mean shift at the output of the layered network, and so it may be removed, as follows.

If we minimise  $E$  with respect to the bias vector only (i.e. differentiate  $E$  with respect to  $|\lambda_0\rangle$  and equate to zero) we find that

$$|\lambda_0\rangle = \overline{|T\rangle} - \mathbf{A}'|m^H\rangle \quad (8)$$

where

$$\overline{|T\rangle} \triangleq \frac{1}{P} \mathbf{T}|1\rangle \quad (9)$$

is the mean target vector with components

$$\overline{T}_i = \frac{1}{P} \sum_{p=1}^P t_{ip} \quad (10)$$

and

$$|m^H\rangle \triangleq \frac{1}{P} \mathbf{H}'|1\rangle \quad (11)$$

is the mean output vector at the hidden units where the  $i$ -th component is

$$\frac{1}{P} \sum_{p=1}^P \phi_i^p \quad (12)$$

Thus, if the square error is minimised, the optimum biases ensure that the mean output of the network equals the mean target pattern.

Substituting equation (8) into equation (7) and then equation (7) into equation (4) implies that the error to be minimised by the weights  $\mathbf{A}'$  may be expressed equivalently as

$$\begin{aligned} E &= \| (\mathbf{T} - \overline{\mathbf{T}}\langle \mathbf{1} |) + \mathbf{A}' (|\mathbf{m}^H\rangle\langle \mathbf{1} | - \mathbf{H}') \|^2 \\ &\equiv \|\hat{\mathbf{T}} - \mathbf{A}'\hat{\mathbf{H}}\|^2 \end{aligned} \quad (13)$$

with the definitions

$$\begin{aligned} \hat{\mathbf{T}} &\triangleq \mathbf{T} - \overline{\mathbf{T}}\langle \mathbf{1} | \\ \hat{\mathbf{H}} &\triangleq \mathbf{H}' - |\mathbf{m}^H\rangle\langle \mathbf{1} | \end{aligned} \quad (14)$$

Thus, we now wish to minimise  $E$  which is a function of scaled targets and inputs, with respect to the parameters  $\lambda_{jk}, \mu_{ij}$ .

It is known [8] that the matrix which minimises the error  $E$  with minimum (Frobenius) norm is the Moore-Penrose pseudo-inverse,  $\hat{\mathbf{H}}^+$ , of matrix  $\hat{\mathbf{H}}$ . The pseudo-inverse  $\mathbf{A}^+$  of the (rectangular) matrix  $\mathbf{A}$  is that unique matrix which satisfies the relationships

$$\begin{aligned} \mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A} \\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+ \\ (\mathbf{A}\mathbf{A}^+)^* &= \mathbf{A}\mathbf{A}^+ \\ (\mathbf{A}^+\mathbf{A})^* &= \mathbf{A}^+\mathbf{A} \end{aligned} \quad (15)$$

In terms of this generalised matrix inverse, the solution of the weight matrix may be expressed as

$$\mathbf{A}' = \hat{\mathbf{T}}\hat{\mathbf{H}}^+ \quad (16)$$

Since the error to be minimised is the trace of the matrix  $(\hat{\mathbf{T}} - \mathbf{A}'\hat{\mathbf{H}})(\hat{\mathbf{T}} - \mathbf{A}'\hat{\mathbf{H}})^*$ , substituting in the solution for  $\mathbf{A}'$  gives, successively,

$$\begin{aligned} E &= \text{Tr} \{ (\hat{\mathbf{T}} - \mathbf{A}'\hat{\mathbf{H}})(\hat{\mathbf{T}} - \mathbf{A}'\hat{\mathbf{H}})^* \} \\ &= \text{Tr} \{ \hat{\mathbf{T}}\hat{\mathbf{T}}^* \\ &\quad - \hat{\mathbf{T}}\hat{\mathbf{H}}^+\hat{\mathbf{H}}\hat{\mathbf{T}}^* - \hat{\mathbf{T}}\hat{\mathbf{H}}^*(\hat{\mathbf{H}}^+)^*\hat{\mathbf{T}}^* \\ &\quad + \hat{\mathbf{T}}\hat{\mathbf{H}}^+\hat{\mathbf{H}}\hat{\mathbf{H}}^*(\hat{\mathbf{H}}^+)^*\hat{\mathbf{T}}^* \} \\ &= \text{Tr} \{ \hat{\mathbf{T}}\hat{\mathbf{T}}^* - \hat{\mathbf{T}}\hat{\mathbf{H}}^*(\hat{\mathbf{H}}\hat{\mathbf{H}}^+)^*\hat{\mathbf{T}}^* \} \end{aligned} \quad (17)$$

To obtain this last relation, we have exploited the properties of the pseudo-inverse given in (15) and the fact that the matrices  $\hat{\mathbf{H}}\hat{\mathbf{H}}^+$  and  $\hat{\mathbf{H}}^+\hat{\mathbf{H}}$  are idempotent.

Note that the matrix  $(\hat{\mathbf{H}}\hat{\mathbf{H}}^*)$  is the *Total Covariance Matrix*,  $\mathbf{S}_T$  at the output of the hidden units,

$$\mathbf{S}_T = \hat{\mathbf{H}}\hat{\mathbf{H}}^* = \sum_{p=1}^P (|\phi^p\rangle - |m^H\rangle) (\langle\phi^p| - \langle m^H|) \quad (18)$$

Thus, since the targets are fixed, minimising  $E$  is equivalent to maximising the cost function

$$C = \text{Tr} \{ \hat{\mathbf{T}}\hat{\mathbf{H}}^*\mathbf{S}_T^+\hat{\mathbf{H}}\hat{\mathbf{T}}^* \} \quad (19)$$

This is the *Network Cost Function*.

If we consider the case where the total covariance matrix is full rank (and so the number of independent patterns is greater than the number of hidden units) then the rank of  $\mathbf{S}_T$  is  $n_0$  and the pseudo-inverse is the true inverse,

$$\mathbf{S}_T^+ \equiv \mathbf{S}_T^{-1} \quad \text{if } \mathbf{S}_T \text{ is full rank.} \quad (20)$$

In a linear problem, this would correspond to an overdetermined situation which does not have a unique solution generally. However, this is usually the case which occurs in

pattern classification tasks where a sufficient number of representative samples need to be considered so that noise effects superimposed on the 'clean' patterns may be compensated for.

We will assume that the total covariance matrix is full rank in what follows, however the actual network cost function (19) is not restricted by this assumption. Under the full rank restriction, it can easily be shown (see Appendix A) that maximising the Network Cost Function is equivalent to maximising the cost function

$$C = \text{Tr} \{ \mathbf{S}_B \mathbf{S}_T^{-1} \} \quad (21)$$

where

$$\mathbf{S}_B \triangleq \hat{\mathbf{H}} \hat{\mathbf{T}}^* \hat{\mathbf{T}} \hat{\mathbf{H}}^* \quad (22)$$

Equations (19) and (21) are the principal results of this paper:

*Minimising the square error at the output of the adaptive layered network, is equivalent to maximising the Network Cost Function (19) at the outputs of the hidden units. If the total covariance matrix is full rank then the Network Cost Function is the trace of the product of a matrix  $\mathbf{S}_B$  and the inverse of the total covariance matrix of the patterns at the outputs of the hidden units (21).*

The following two subsections provide an interpretation of matrix  $\mathbf{S}_B$  for specific choices of the output coding for the target vectors.

### 3.1 Particular target coding schemes.

Consider the specific choice of a one-from- $n'$  coding scheme. Along with the other assumptions made on the form of the adaptive layered network structure, the desired target value of a particular pattern is unity if the chosen input pattern is in that class, and is zero otherwise. If there are  $n'$  classes,  $C_k, k = 1, \dots, n'$  with  $n_k$  patterns in class  $C_k$ , then for

this particular coding scheme, the matrix  $S_B$  introduced in the previous section may be expanded as

$$S_B = \hat{H} \hat{T}^* \hat{T} \hat{H}^* \quad (23)$$

$$= \sum_{k=1}^{n'} n_k^2 \left( |m_k^H\rangle - |m^H\rangle \right) \left( \langle m_k^H| - \langle m^H| \right)$$

where  $|m_k^H\rangle$  is the mean output vector over all patterns in class  $C_k$ ,

$$|m_k^H\rangle = \frac{1}{n_k} \sum_{|\phi^p\rangle \in C_k} |\phi^p\rangle \quad (24)$$

This equation is recognised as the expression for the *weighted* between class covariance matrix. Thus, for a one-from- $n'$  output coding (which is very common in the literature) the layered network maximises a cost function which is the trace of the product of the weighted between class covariance matrix, and the inverse of the total covariance matrix. This is an interesting result, since it illustrates how adaptive layered networks implicitly incorporate the proportions of samples within each class as priors.

Consider an alternative coding scheme: the target of an input pattern is zero if the pattern is not in the class under consideration and is the reciprocal of the square root of the number of patterns in that class otherwise.

$$t_{kp} = \begin{cases} 1/\sqrt{n_k} & \text{if } |\phi^p\rangle \in C_k \\ 0 & \text{otherwise} \end{cases}$$

In this case, the matrix  $S_B$  expands to

$$\tilde{S}_B = \sum_{k=1}^{n'} n_k \left( |m_k^H\rangle - |m^H\rangle \right) \left( \langle m_k^H| - \langle m^H| \right) \quad (25)$$

which is the conventional (*not weighted by priors*) between class covariance matrix. Thus, in a pattern classification problem which would be solved best by producing a pattern distinction which *uniformly* weights the classes, an adaptive layered network trained on a one-from- $n'$  coding scheme would not produce the best results. For instance, in modelling

continuous speech patterns the bulk of the acoustic vectors may represent silence. To ensure that silence did not dominate the classification performance, but instead concentrated on the more relevant information-bearing acoustic vectors, the adaptive layered network itself should not incorporate prior knowledge on the frequency of occurrence of patterns within each class for the between class covariance matrix. In order to force the network weights not to bias in favour of the classes with largest membership, the prior knowledge of pattern distribution has to be encoded in the target vectors. In such experiments, unevenly distributed training patterns between classes may be compensated for by scaling the 1-from- $n'$  target vectors by the square root of the number in each class.

Two particular instances where the distinction between the weighted and not-weighted between class covariance matrices will not be made occur when the number of patterns in each class is the same and in a two-class problem. In this latter case, the weighted between class covariance matrix,  $S_B$ , and the conventional between class covariance matrix,  $\tilde{S}_B$ , are connected by a multiplicative constant.

$$S_B = \frac{2n_1n_2}{P} \tilde{S}_B$$

Thus, maximising the cost function with the weighted covariance will give the same result as maximising with the conventional covariance matrix for a two class problem.

### 3.2 The Linear Adaptive Layered Network.

The final special case to consider is when the hidden units are constrained to a linear transfer function of the fan-in ( $\phi_j(x) = x \forall j$ ). In this case, the adaptive layered network as a whole performs a linear transformation between the input and output spaces. This was the situation considered theoretically by Gallinari *et.al.* [4]. The result which they proved (under certain reasonable assumptions) was that the weight matrix between the input and hidden units which minimised the square error of the network, also maximised the cost function

$$\tilde{C}(W') = \frac{|W' \cdot S_B^I \cdot W'|}{|W' \cdot S_T^I \cdot W'|} \quad (26)$$

which is the ratio of the determinants of the between class, and total covariance matrices in the transformed space of the input patterns. In this equation,  $S_B^I$  and  $S_T^I$  are the between

class and total covariance matrices of the original input data. It was not clear what other types of cost function would be maximised also by the same matrix  $W'$ , although the above choice is a reasonable one to make for discrimination analysis. However, this choice is *not* the natural cost function which the network is implicitly attempting to maximise, as we have illustrated.

It is interesting to see if the result presented in equation (26) can reproduce the maximisation of  $C(W')$  (21) in the limit of a linear network. It will be shown that this is indeed the case. The first part of the illustration removes the effects of the biases by demonstrating that the hidden unit bias vector compensates for the difference between the mean vector over all patterns at the output of the hidden units and the transform of the mean of all the input patterns. The second part demonstrates that the matrix equation satisfied by the weights which maximises the cost function,  $C$ , is the almost the same as the equation for the matrix which maximises the cost function  $\tilde{C}$ . Thus, the explicit solution for the weights which maximises  $\tilde{C}$  will also maximise  $C$ .

If the network is linear, then the output matrix of the hidden units,  $H'$ , which is of size  $n_0 \times P$ , is obtained by a linear transformation of the  $(n+1) \times P$  array of input patterns,  $I$  by the  $n_0 \times (n+1)$  weight matrix  $W$  (note that  $W$  and  $I$  include the effect of the biases,  $|\mu_0\rangle$ ),

$$H' = WI$$

As illustrated previously, the effect of the biases may be separated by decomposing the matrix  $H'$  into

$$H' = W'I' + |\mu_0\rangle\langle 1| \quad (27)$$

where  $W'$  is of dimension  $n_0 \times n$ ,  $I'$  is of dimension  $n \times P$ ,  $|\mu_0\rangle$  is an  $n_0$  dimensional column vector and  $\langle 1|$  is an  $n_0$  dimensional unit constant row vector.

Since  $H'|1\rangle = P|m^H\rangle$ , using the above equation gives the result that

$$|\mu_0\rangle = |m^H\rangle - W'|m^I\rangle \quad (28)$$

where  $|m^I\rangle$  is the mean input pattern.

Thus, in the case of linear hidden units, the hidden unit bias vector is to compensate for the difference between the actual mean of the output patterns of the hidden units, and the linear transformation of the mean of the input patterns.

Substituting back for  $\langle \mu_0 \rangle$  in (27) allows the bias to be removed by considering mean-shifted input patterns and hidden unit outputs:-

$$\begin{aligned} \mathbf{H}' - \langle \mathbf{m}^H \rangle \langle \mathbf{1} | &= \mathbf{W}' \mathbf{I}' - \mathbf{W}' \langle \mathbf{m}^I \rangle \langle \mathbf{1} | \\ &\equiv \hat{\mathbf{H}} = \mathbf{W}' \hat{\mathbf{I}} \end{aligned} \quad (29)$$

where

$$\begin{aligned} \hat{\mathbf{H}} &\triangleq \mathbf{H}' - \langle \mathbf{m}^H \rangle \langle \mathbf{1} | \\ \hat{\mathbf{I}} &\triangleq \mathbf{I}' - \langle \mathbf{m}^I \rangle \langle \mathbf{1} | \end{aligned} \quad (30)$$

In terms of the rescaled input patterns and hidden unit outputs, the matrices  $\mathbf{S}_B$  and  $\mathbf{S}_T$  may be expressed as

$$\begin{aligned} \mathbf{S}_B &= \mathbf{W}' \hat{\mathbf{I}} \hat{\mathbf{I}}' \mathbf{W}' \\ &= \mathbf{W}' \mathbf{S}_B^I \mathbf{W}' \end{aligned} \quad (31)$$

$$\begin{aligned} \mathbf{S}_T &= \mathbf{W}' \hat{\mathbf{H}} \hat{\mathbf{H}}' \mathbf{W}' \\ &= \mathbf{W}' \mathbf{S}_T^I \mathbf{W}' \end{aligned} \quad (32)$$

where  $\mathbf{S}_B^I$  and  $\mathbf{S}_T^I$  are the 'between class' (provided the targets are appropriately chosen) and total covariance matrices of the input patterns. Thus, in terms of these matrices associated with the input data, the cost function that the network is attempting to maximise is

$$C = \text{Tr} \left\{ \mathbf{W}' \mathbf{S}_B^I \mathbf{W}' \left( \mathbf{W}' \mathbf{S}_T^I \mathbf{W}' \right)^{-1} \right\} \quad (33)$$

which is performed by an appropriate choice of the weight matrix  $\mathbf{W}'$  (the links between the input and hidden units).

The next part of this illustration is to show that any matrix which maximises the cost function

$$\tilde{C}(W') = \frac{|W'^S S_B^I W'|}{|W'^S S_T^I W'|} \quad (34)$$

also maximises the cost function  $C(W')$  (33).

Taking the derivative of the cost function  $C(W')$  with respect to the elements of matrix  $W'$  and equating to zero, gives the matrix equation

$$2 (W'^S S_T^I W'^S)^{-1} [W'^S S_B^I - W'^S S_B^I W'^S (W'^S S_T^I W'^S)^{-1} W'^S S_T^I] = 0 \quad (35)$$

Similarly, the derivative of the cost function  $\tilde{C}(W')$  with respect to  $W'$  equated to zero gives the matrix equation:-

$$2 \frac{|W'^S S_B^I W'^S|}{|W'^S S_T^I W'^S|} (W'^S S_B^I W'^S)^{-1} [W'^S S_B^I - W'^S S_B^I W'^S (W'^S S_T^I W'^S)^{-1} W'^S S_T^I] = 0 \quad (36)$$

The derivation of this equation has assumed that the rank of  $S_B^I$  is at least  $n_0$  so that the inverse (and hence the derivative of the determinant) exists.

By comparing equations (35), and (36), it is evident that a nontrivial solution for matrix  $W'$  which satisfies (36) must also satisfy (35). In particular, any  $W'$  which satisfies the generalised eigenvalue equation,

$$W'^S S_B^I = \Gamma W'^S S_T^I \quad (37)$$

where  $\Gamma$  is a diagonal matrix of eigenvalues, satisfies (36) and maximises the cost function  $\tilde{C}$  with a value of  $|\Gamma|$  which is the product of the eigenvalues. This solution also satisfies (35) and maximises the Network Cost Function,  $C$  with a value of  $Tr \Gamma$  which is the sum of the eigenvalues.

Thus  $W'$  may be composed out of the eigenvectors of  $S_B^I (S_T^I)^{-1}$  corresponding to the non zero eigenvalues (giving a specific solution  $W_0$ ). Note that any linear, invertible transformation,  $V$ , of  $W_0$ , will also maximise these two cost functions; hence the solution is not unique.

However, there exist solutions of (36) which do *not* maximise the cost function  $\tilde{C}$  but which do still maximise the Network Cost Function  $C$ . For instance, the matrix  $\tilde{W} = VW_0 + M$  where the columns of  $M$  lie in the null space of  $S_B^T$  (so that  $S_B^T M = 0$ ) still satisfies the error minimisation equations, but it does not maximise the cost function  $\tilde{C}$ . However, the Network Cost Function is not altered by the addition of such a null subspace matrix. Therefore, there exist solutions which minimise the network error and maximise the network cost function, but which do not maximise the cost function proposed by Gallinari *et.al.* Note that if a *minimum norm* solution is demanded, then matrix  $W'$  must lie entirely within the image of  $S_B^T$  and both cost functions are maximised simultaneously.

#### 4 Numerical Illustration.

In this section, the implications of the theorem in Section 3 are illustrated. The problem is to determine the number of connected groups of 1's in an 8-bit binary string (the 'Contiguity Problem'). Thus, there are 256 distinct patterns, each of which belongs to one of 5 classes. Table 1 gives the number of members of each class. Note that this is not a 'typical' problem; there is no true concept of noise in the data and, being a Boolean problem, it does not really make sense to discuss its statistics in terms of covariance matrices. Nevertheless, it is a simple problem with more than two classes to discriminate and with a disparate number of patterns in each class. Despite the lack of an intuitive interpretation of covariance matrices for such a problem, it should still be true that the Network Cost Function is maximised - the issue is not what value the Network Cost Function attains (which will presumably be larger for Gaussian distributed input patterns) but whether the value that it does reach is the largest possible value consistent with the problem and the network. Thus we have chosen this problem to be *illustrative* of the general results we have derived.

The specific nonlinear transformation from the input layer to the hidden layer employed in this test is that transformation as determined by a multi-layer perceptron with 8 input units, 5 output units and a varying number of hidden units as the classification network. The output units have a linear transfer function. The output coding scheme adopted is a one-from-five coding, so that the matrix  $S_B$  is given by equation (23). Since there are an

Class	Number of connected groups	Number of members in class
1	0	1
2	1	36
3	2	128
4	3	84
5	4	9

Table 1: Numbers of members in each class for the connected groups of digit 1's

unequal number of members in each class, this matrix is not proportional to the conventional between class covariance matrix,  $\tilde{S}_B$  (25).

Four cost functions were evaluated at the output of the hidden units, namely the Network Cost Function,  $C = \text{Tr}(\mathbf{S}_B \mathbf{S}_T^{-1})$ ; the cost function,  $C = \text{Tr}(\tilde{\mathbf{S}}_B \mathbf{S}_T^{-1})$ ; and the ratios of determinants  $|\mathbf{S}_B|/|\mathbf{S}_T|$  and  $|\tilde{\mathbf{S}}_B|/|\mathbf{S}_T|$ . The method of solution of the least squares problem uses an iterative scheme to minimise the error (see Appendix B) and the cost functions are evaluated at each stage of the iteration.

Although this is a rather artificial problem, it is one which can not be solved by a linear transformation from input space to output space, or equivalently, a network with four linear transfer functions at the hidden layer [4]. Performing a least-means-squares mapping from the 8 input units directly to the 5 output units, and classifying the patterns according to the minimum Euclidean distance in the output space gives 132 (= 51.56%) correct solutions (and one indeterminate solution since the null vector maps on to the null vector, which is equi-distant from all classes). A network with four (linear) hidden units achieves the same performance, though the addition of the biases does enable the null vector to be classified correctly. Figure 1 plots the cost functions as a function of iteration number in the error minimisation routine (in fact, for this example, we have chosen to use an inefficient algorithm - steepest descents - to solve for the parameters of the network, since the BFGS routine used in the nonlinear problems converged in too few iterations to illustrate the problem). The figure shows that both the trace cost functions reach a maximum at the end of the iteration whilst the ratio of determinants cost function reaches a maximum after about 20 iterations and then starts to decline. This is because the solution for the matrix of weights

connecting the input units to the hidden units is not a minimum norm solution since it has components in the null space of the between class covariance matrix  $\tilde{S}_B$ .

Introducing a nonlinear transfer function at each of the hidden units gives improved classification performance. Figure 2 plots the cost functions as a function of iteration number for a network with four hidden units. For the particular random start configuration of the weights and biases chosen, the network achieved 189 (73.83%) correct solutions. The network cost function increases monotonically with the number of iterations of the algorithm. The sum-squared error at the output decreases monotonically correspondingly. However, the cost function  $C = \text{Tr}(\tilde{S}_B S_T^{-1})$  settles at a value which is not its peak value during the iteration. Both cost functions which depict the ratio of determinants of the between class and total covariance matrices are *not* maximised. In fact, in the situation where the number of hidden units is equal to one less than the number of classes ( $n_0 = n' - 1$ ), as illustrated in this example, the determinants  $|S_B|$  and  $|\tilde{S}_B|$  are related by

$$|S_B| = n' \frac{\prod_{i=1}^{n'} n_i}{\sum_{i=1}^{n'} n_i} |\tilde{S}_B| \quad (38)$$

Thus both cost functions exhibit the same behaviour, as observed in the figures.

With a nonlinear network, we are not restricted to having fewer hidden units than the number of classes as in linear discriminant analysis and Figure 3 plots the cost functions as a function of iteration number for a network with 6 hidden units. With this number of hidden units, the determinant of the between-class covariance matrix is identically equal to zero since the dimension of the matrix  $S_B$  is greater than the number of classes. Therefore, it is not meaningful to use the ratio-of-determinants cost function as a measure of classification performance. Consequently, an additional cost function,  $\text{Tr}\{\tilde{S}_B\}/\text{Tr}\{S_T\}$ , the ratio of traces of the between class and total covariance matrices has also been plotted for comparison. Note that for the particular random start configuration used for this figure, the network achieved 192 (= 75%) correctly classified solutions.

This figure shows that the Network Cost Function is maximised as the error is minimised but the trace of the product of the conventional between class covariance matrix and the inverse of the total covariance matrix is not maximised. For a larger number of hidden units we find, for this particular problem, that the matrix  $S_T$  becomes singular during the

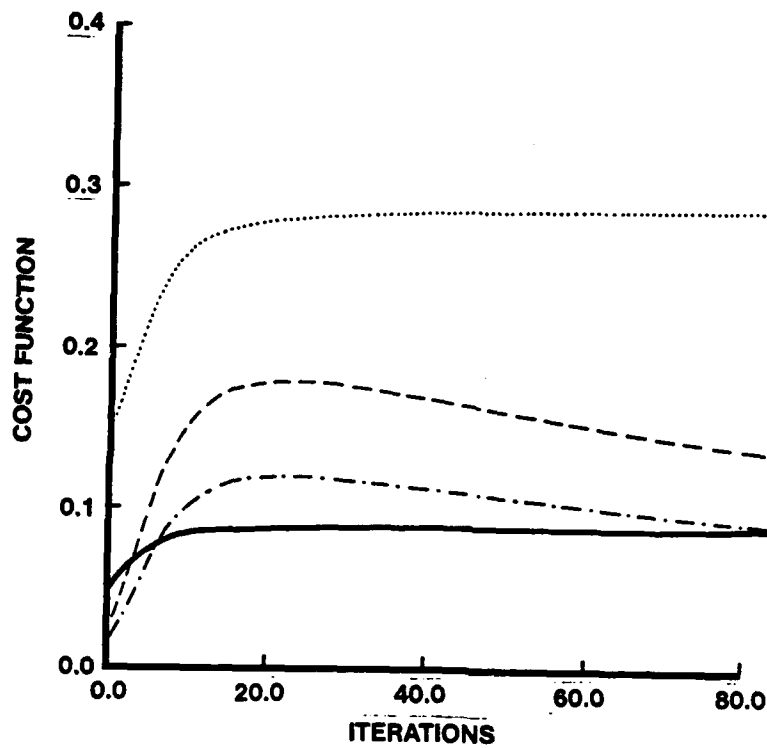


Figure 1: Plots of the network cost function  $Tr(S_B S_T^{-1})$  divided by  $||\hat{T}||^2$  (solid line), the cost function  $Tr(\hat{S}_B S_T^{-1})$  divided by  $||\hat{T}||^2$  (dotted line); the ratio of determinants  $|S_B|/|S_T|$  multiplied by 10 (dot-dash), and the ratio of determinants  $|\hat{S}_B|/|S_T|$  multiplied by  $10^6$  (dashed line) as a function of iteration number in a least-squares minimisation routine, for a network with four hidden units with linear transfer functions.

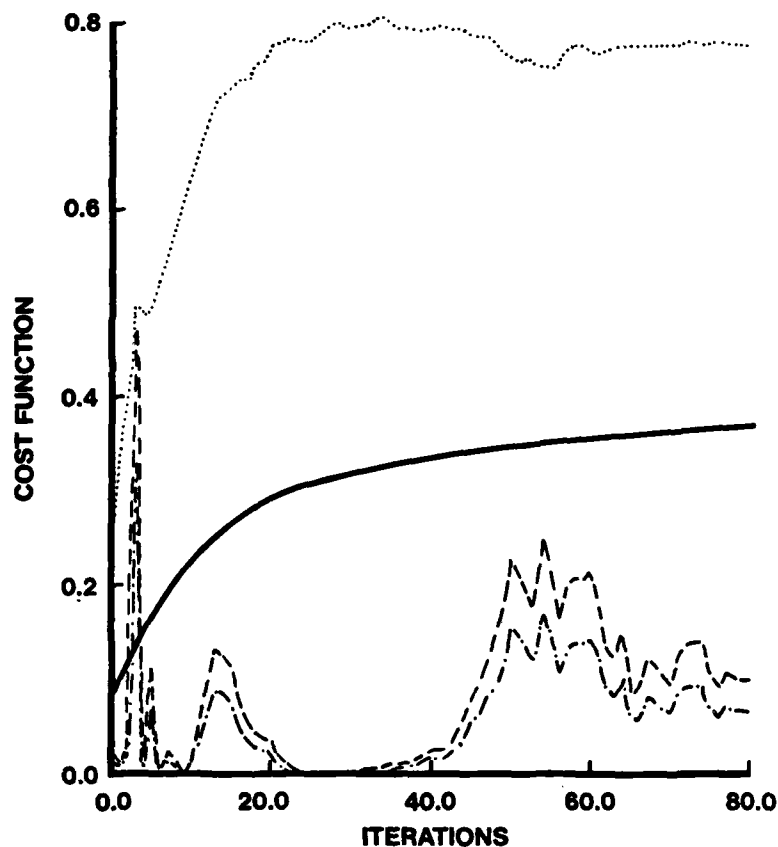


Figure 2: Plots of the network cost function  $\text{Tr}(\mathbf{S}_B \mathbf{S}_T^{-1})$  divided by  $\|\hat{\mathbf{T}}\|^2$  (solid line), the cost function  $\text{Tr}(\hat{\mathbf{S}}_B \mathbf{S}_T^{-1})$  divided by  $\|\hat{\mathbf{T}}\|^2$  (dotted line); the ratio of determinants  $|\mathbf{S}_B|/|\mathbf{S}_T|$  multiplied by 10 (dot-dash), and the ratio of determinants  $|\hat{\mathbf{S}}_B|/|\mathbf{S}_T|$  multiplied by  $10^6$  (dashed line) as a function of iteration number in a least-squares minimisation routine, for a network with four hidden units with nonlinear transfer functions.

iteration and therefore the more general form of the network cost function (19) must be used.

These three figures illustrate a natural evolution of discriminant analysis strategies. The linear network produces an optimum linear transformation to a dimension reducing subspace where the patterns corresponding to different classes are in some sense maximally separated, and the patterns within each class are grouped (this is the example illustrated by Figure 1). The next step is to allow for a nonlinear transformation on to a dimension reducing subspace which should have the advantage of providing a better class discrimination transformation (the example illustrated by Figure 2). The final stage (Figure 3) is to allow for an embedding of the input patterns by a nonlinear transformation to a higher dimensional space where an even better class separation may be achieved. Once a transformation has been performed into a space where the transformed patterns are more easily distinguished, it is much easier for a linear discrimination (the hidden-output layer of the multi-layer perceptron considered in the paper) to perform good classification. These general comments are reflected in the classification performance of the figures which rises from 132 to 192 patterns classified correctly. However, note that in all instances, the criterion for maximal class separation is determined by the Network Cost Function. This may not be the best criterion to choose for a general discrimination problem, but it is the only one that such an adaptive feed-forward layered network can employ.

## 5 Conclusion.

Adaptive feed-forward layered networks are capable of performing classification tasks better than traditional methods. This paper has demonstrated that this ability arises out of the implicit way in which a Network Cost Function is maximised in the space of the hidden units.

Specifically, this paper considered a general nonlinear transformation from the input patterns to a set of patterns in the space defined by the final layer of hidden units (there is no restriction on the number of layers constituting the nonlinear transformation) followed by a linear transformation to a set of output target patterns. If the network weights are

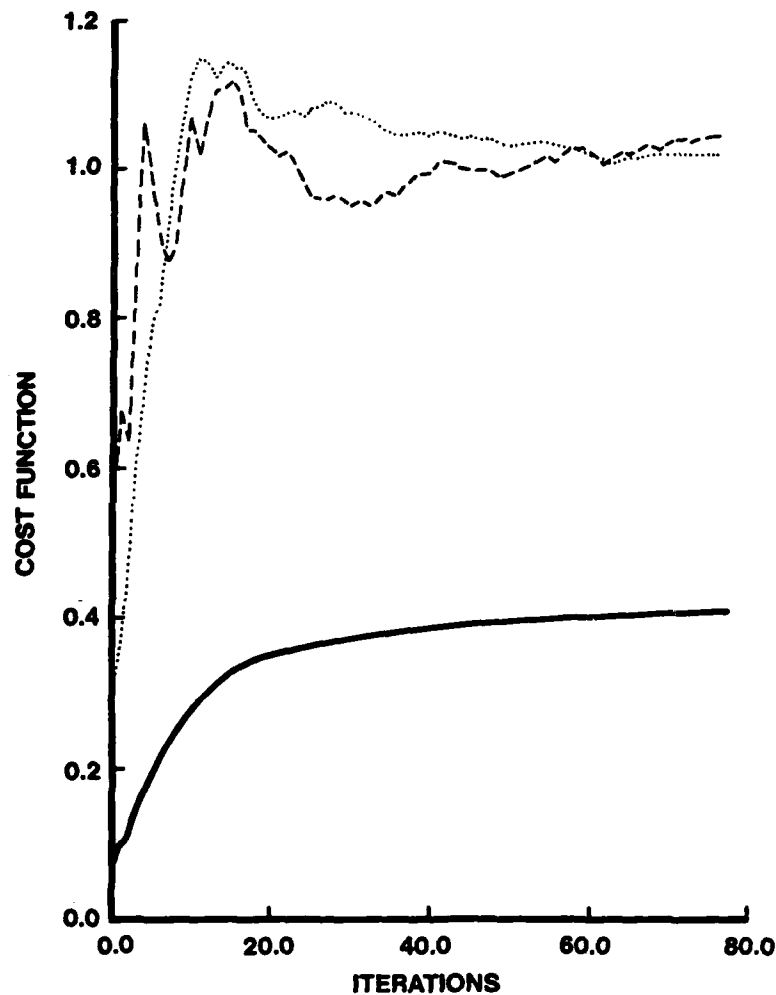


Figure 3: Plots of the network cost function  $Tr(\mathbf{S}_B \mathbf{S}_T^{-1})$  divided by  $\|\hat{\mathbf{T}}\|^2$  (solid line), the cost function  $Tr(\hat{\mathbf{S}}_B \mathbf{S}_T^{-1})$  divided by  $\|\hat{\mathbf{T}}\|^2$  (dotted line) and the ratio of traces  $Tr(\hat{\mathbf{S}}_B)/Tr(\mathbf{S}_T)$  multiplied by a factor of 10 (dashed line) as a function of iteration number in a least-squares minimisation routine, for a network with six hidden units with nonlinear transfer functions.

adjusted to minimise the mean-square error between the desired target patterns and the actual output patterns of the network, then this is equivalent to maximising the Network Cost Function

$$C = \text{Tr} \{ \hat{T} \hat{H}^* S_T^* \hat{H} \hat{T} \} \quad (39)$$

where  $\hat{T}$  and  $\hat{H}$  are defined in equation (14) and  $S_T$  is the total covariance matrix of the patterns at the outputs of the final hidden layer.

If this total covariance matrix is full rank (which is usually the case) maximising the Network Cost Function is equivalent to maximising the cost function

$$C = \text{Tr} \{ S_B S_T^{-1} \} \quad (40)$$

The matrix  $S_B$  may be interpreted to be the weighted between class covariance matrix at the output of the final hidden layer if the target patterns are chosen as a 1-from- $n'$  coding. Equivalently, encoding the distribution of patterns between the classes into the target patterns (which is equivalent to weighting the error minimisation) allows the matrix  $S_B$  to be interpreted as the conventional between class covariance matrix.

The action of a feed-forward network does not maximise more traditional cost functions employed in discrimination analysis, as our numerical example illustrated. However in the special case of a totally linear network with one hidden layer, the minimum norm solution which maximises the ratio of determinants of the between class and total covariance matrices (the result obtained by Gallinari *et.al* [4]) is equivalent to maximising the Network Cost Function.

Thus an adaptive feed-forward layered network performs a natural generalisation of linear discriminant analysis by *implicitly* maximising a cost function relating the between class and total covariance matrices. This is precisely why such networks have been demonstrated to perform classification tasks well. It should be possible to force such networks to maximise alternative network cost functions more appropriate to a specific task, by minimising different error measures to the mean-square-error considered in this paper. Alternatively, given a cost function which it is desired to maximise explicitly, what error function should be minimised by first performing a linear transformation to a classification space which

would implicitly achieve the same effect. In this sense, the results of this paper may have a wider applicability in 'designer networks' for specific applications.

## References

- [1] Paul Gorman, R., Sejnowski, Terrence J.,(1988). Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets, *Neural Networks*, 1, 75-89.
- [2] Webb, A.R., Lowe, David, Bedworth, M.D.,(1988). A Comparison of Nonlinear Optimisation Strategies for Feed-Forward Adaptive Layered Networks., *Memorandum 4157, Royal Signals and Radar Establishment, St Andrews Rd., Great Malvern, Worcestershire, WR14 3PS, U.K..*
- [3] Boulard, H., Kamp, Y.,(1987). Auto-association by Multilayer Perceptrons and Singular Value Decomposition, *Manuscript M217, Philips Research Laboratory, Av. Van Becelaere 2-Box 8, B-1170 Brussels, Belgium.*
- [4] Gallinari, P., Thiria, S., Fogelman Soulie, F.,(1988). Multilayer Perceptrons and Data Analysis, *IEEE Annual International Conference on Neural Networks, San Diego, California, I, 1-391-1-399.*
- [5] Broomhead, D.S., Lowe, David,(1988). Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks, *Memorandum 4148, Royal Signals and Radar Establishment, St Andrews Rd., Great Malvern, Worcestershire, WR14 3PS, U.K..*
- [6] Devijver, P.A., Kittler, J.,(1982). *Pattern Recognition: A Statistical Approach*, Prentice-Hall International.
- [7] Webb, A.R., Lowe, David, (1988). A Hybrid Optimisation Strategy for Adaptive Feed-forward Layered Networks, *Memorandum 4193, Royal Signals and Radar Establishment, St Andrews Rd., Great Malvern, Worcestershire, WR14 3PS, U.K..*
- [8] Golub, G., Kahan, W.,(1965). Calculating the Singular Values and Pseudo-Inverse of a Matrix, *Journal SIAM Numerical Analysis, Series B*, 2(2), 205-224.

## A Proof of the cost function equation.

In section 3, it was shown how minimising the error was equivalent to maximising the cost function in equation (19), i.e.

$$\begin{aligned} C &= \text{Tr} \{ \hat{T} \hat{H}^* S_T^+ \hat{H} \hat{T}^* \} \\ &= \text{Tr} D \end{aligned} \quad (41)$$

If we consider the eigenvalue equation of matrix  $D$

$$D|x\rangle = \eta|x\rangle \quad (42)$$

and operate through the equation on the left by the matrix  $\hat{H} \hat{T}^*$ , one finds

$$(\hat{H} \hat{T}^* \hat{T} \hat{H}^*) S_T^+ |y\rangle = \eta |y\rangle \quad (43)$$

This is an eigenvalue equation for the matrix

$$E = \hat{H} \hat{T}^* \hat{T} \hat{H}^* S_T^+$$

with different eigenfunctions

$$|y\rangle = \hat{H} \hat{T}^* |x\rangle.$$

All the eigenvalues of  $D$  are also eigenvalues of  $E$  (but note that  $E$  may have additional eigenvalues to  $D$ ). Since the trace of a matrix is the sum of its eigenvalues, then the trace of  $D$  is less than or equal to the trace of matrix  $E$ . In the case of a full rank total covariance matrix,  $S_T$ , then  $E$  has the same rank as  $D$ . Thus,  $E$  has the same number of eigenvalues of  $D$  and so  $\text{Tr} D = \text{Tr} E$ . Consequently, maximising the Network Cost Function is equivalent to maximising the cost function given in equation (21), as desired. ■

Provided that the total covariance matrix is full rank, then the pseudo-inverse equals the true inverse and the above conclusions may be reversed: *maximising the cost function (21) is equivalent to maximising the cost function (19)*. Thus either of the cost functions (19), (21) may be taken to be the Network Cost Function. This is often the case in practice. Unfortunately, this conclusion does not apply if the total covariance matrix  $S_T$  is not full rank since there will exist eigenvalues of  $E$  which are not eigenvalues of  $D$  and so  $\text{Tr} E \geq \text{Tr} D$ .

## B Numerical Solution of the Least-squares Problem.

In the numerical example used to illustrate the theorem, the network employed had a single hidden layer with the output nonlinearity of the hidden units described by a logistic function,

$$\phi(x) = \frac{1}{1 + \exp(-x)} \quad (44)$$

and an output layer employing linear units.

The square error at the output of the network is regarded as a nonlinear function of the weights and biases between the input layer and the hidden layer. This error may be minimised using any suitable nonlinear function optimisation strategy [2], and we have chosen to use a quasi-Newton technique, the BFGS method.

The minimisation proceeds as follows. Given an initial estimate,  $\{\mu(t=0)\}$ , of the weights and biases  $\{\mu\}$  (chosen from a uniform random distribution in the interval  $(-1, 1)$ ) between the input and hidden layers, the final layer weights and biases,  $\{\lambda\}$ , are calculated using equations (16) and (8) and the value of the output error obtained. The gradient of the error with respect to the parameters  $\{\mu\}$  may then be calculated from Equation (3).

Thus, given an initial position and an initial search direction (taken to be the direction of the downhill gradient), the algorithm performs a search along this direction to obtain an estimate of the minimum of the error in this direction. Once this has been achieved, a new search direction is generated (using the BFGS prescription) and a search performed to find the minimum of the error in this new direction. This procedure continues until convergence. Note that each time that the error is evaluated (for each new estimate of the parameters  $\{\mu\}$ ) the values of the parameters  $\{\lambda\}$  must be obtained using equations (16) and (8) prior to evaluation of the error. In this way, the values of the parameters  $\{\lambda\}$  are tied to the values of  $\{\mu\}$ . This ensures that the method produces a global minimum in the subspace spanned by the parameters  $\{\lambda\}$ .


Note that the search strategy to find a minimum of the nonlinear function, could have been performed by a standard (accelerated) steepest descents procedure. In our experience [2], this would have taken at least an order of magnitude longer in terms of CPU time,

or the number of iterations. It was decided that the BFGS procedure was one of the more efficient techniques to use for this size of problem.

## DOCUMENT CONTROL SHEET

Overall security classification of sheet ...UNCLASSIFIED.....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S) )

1. DRIC Reference (if known)	2. Originator's Reference Memorandum 4209	3. Agency Reference	4. Report Security Classification Unclassified	
5. Originator's Code (if known) 7784000	6. Originator (Corporate Author) Name and Location Royal Signals and Radar Establishment St Andrews Road, Malvern, Worcestershire WR14 3PS			
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location			
7. Title A THEOREM CONNECTING ADAPTIVE FEED-FORWARD LAYERED NETWORKS AND NONLINEAR DISCRIMINANT ANALYSIS				
7a. Title in Foreign Language (in the case of translations)				
7b. Presented at (for conference papers) Title, place and date of conference				
8. Author 1 Surname, initials Webb A R	9(a) Author 2 Lowe D	9(b) Authors 3,4...	10. Date 8.88	pp. ref. 26
11. Contract Number	12. Period	13. Project	14. Other Reference	
15. Distribution statement Unlimited				
Descriptors (or keywords)				
 continue on separate piece of paper				
<b>Abstract</b> This Memorandum provides a theorem which illustrates why a general adaptive feed-forward layered network with linear output units can perform well as a pattern classification device. The central result is that minimising the error at the output of the network is equivalent to maximising a particular norm, the Network Cost Function, at the output of the hidden units. If the total covariance matrix is full rank and the targets are appropriately chosen, then this cost function relates the inverse of the total covariance matrix and the weighted between class covariance matrix of the hidden unit patterns. In a linear network it is shown how our theorem can reproduce the result recently obtained by Gallinari et al as a special case. We present numerical simulations to illustrate the theorem and to show that alternative choices for the cost function at the hidden layer are not maximised, generally, in a nonlinear situation.				

S80/48

(2401 14.21714) - (24) + 1